Lecture 21a: Collinearity & Outliers

1 Why Collinearity Is a Problem

Remember our formula for the estimated coefficients in a multiple linear regression:

$$\widehat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$$

This is obviously going to lead to problems if $\mathbf{X}^T\mathbf{X}$ isn't invertible. Similarly, the variance of the estimates,

$$\mathrm{Var}\left[\widehat{\beta}\right] = \sigma^2(\mathbf{X}^T\mathbf{X})^{-1}$$

will blow up when $\mathbf{X}^T\mathbf{X}$ is singular. If that matrix isn't exactly singular, but is close to being non-invertible, the variances will become huge.

There are several equivalent conditions for any square matrix $\mathbf{U}$ to be singular or non-invertible:

- The determinant $\det \mathbf{U}$ (or $|\mathbf{U}|$) is 0.
- At least one eigenvalue of u is 0. (This is because the determinant of a matrix is the product of its eigenvalues.)
- $\mathbf{U}$ is **rank deficient**, meaning that one or more of its columns (or rows) is equal to a linear combination of the other rows.

Since we're not concerned with any old square matrix, but specifically with $\mathbf{X}^T\mathbf{X}$, we have an additional equivalent condition:

- $\mathbf{X}$ is **column-rank** deficient, meaning one or more of its columns is equal to a linear combination of the others.

The last explains why we call this problem **collinearity**: it looks like we have p different predictor variables, but really some of them are linear combinations of the others, so they don't add any information. If the exact linear relationship holds among more than two variables, we talk about **multicollinearity**; **collinearity** can refer either to the general situation of a linear dependence among the predictors, or, by contrast to multicollinearity, a linear relationship among just two of the predictors.

Again, if there isn't an *exact* linear relationship among the predictors, but they're close to one, $\mathbf{X}^T\mathbf{X}$ will be invertible, but $(\mathbf{X}^T\mathbf{X})^{-1}$ will be huge, and the variances of the estimated coefficients will be enormous. This can make it very hard to say anything at all precise about the coefficients, but that's not *necessarily* a problem.

1.1 Dealing with Collinearity by Deleting Variables

Since not all of the p variables are actually contributing information, a natural way of dealing with collinearity is to drop some variables from the model. If you want to do this, you should think very carefully about *which* variable to delete. As a concrete example: if we try to include all of a student's grades as predictors, as well as their over-all GPA, we'll have a problem with collinearity (since GPA is a linear function of the grades). But depending on what we want to predict, it might make more sense to use just the GPA, dropping all the individual grades, or to include the individual grades and drop the average.

1.2 Diagnosing Collinearity Among Pairs of Variables

Linear relationships between pairs of variables are fairly easy to diagnose: we make the pairs plot of all the variables, and we see if any of them fall on a straight line, or close to one. Unless the number of variables is huge, this is by far the best method. If the number of variables *is* huge, look at the correlation matrix, and worry about any entry off the diagonal which is (nearly) ± 1.

1.3 Why Multicollinearity Is Hard to Detect

A multicollinear relationship involving three or more variables might be totally invisible on a pairs plot. For instance, suppose X_1 and X_2 are independent Gaussians, of equal variance σ^2, and X_3 is their average, $X_3 = (X_1 + X_2)/2$. The correlation between X_1 and X_3 is

$$\begin{aligned} \mathrm{Cor}(X_1, X_3) &= \frac{\mathrm{Cov}\left[X_1, X_3\right]}{\sqrt{\mathrm{Var}\left[X_1\right]\mathrm{Var}\left[X_3\right]}} && (1) \\ &= \frac{\mathrm{Cov}\left[X_1, (X_1+X_2)/2\right]}{\sqrt{\sigma^2\sigma^2/2}} = \frac{\sigma^2/2}{\sigma^2/\sqrt{2}} = \frac{1}{\sqrt{2}}. && (2) \end{aligned}$$

This is also the correlation between X_2 and X_3. A correlation of $1/\sqrt{2}$ isn't trivial, but is hardly perfect, and doesn't really distinguish itself on a pairs plot (Figure 1).

```
x1 = rnorm(100,70,15)
x2 = rnorm(100,70,15)
x3 = (x1 + x2)/2
X  = cbind(x1,x2,x3)
pairs(X)
cor(X)
```

```
##            x1         x2        x3
## x1 1.00000000 0.03788452 0.7250514
## x2 0.03788452 1.00000000 0.7156686
## x3 0.72505136 0.71566863 1.0000000
```

2 Variance Inflation Factors

If the predictors are correlated with each other, the standard errors of the coefficient estimates will be bigger than if the predictors were uncorrelated. If the predictors were uncorrelated, the variance of $\widehat{\beta}_i$ would be

$$\mathrm{Var}\left[\widehat{\beta}_i\right] = \frac{\sigma^2}{n s^2_{X_i}} \quad (3)$$

just as it is in a simple linear regression. With correlated predictors, however, we have to use our general formula for the least squares:

$$\mathrm{Var}\left[\widehat{\beta}_i\right] = \sigma^2 (\mathbf{X}^T\mathbf{X})^{-1}_{i+1,i+1} \quad (4)$$

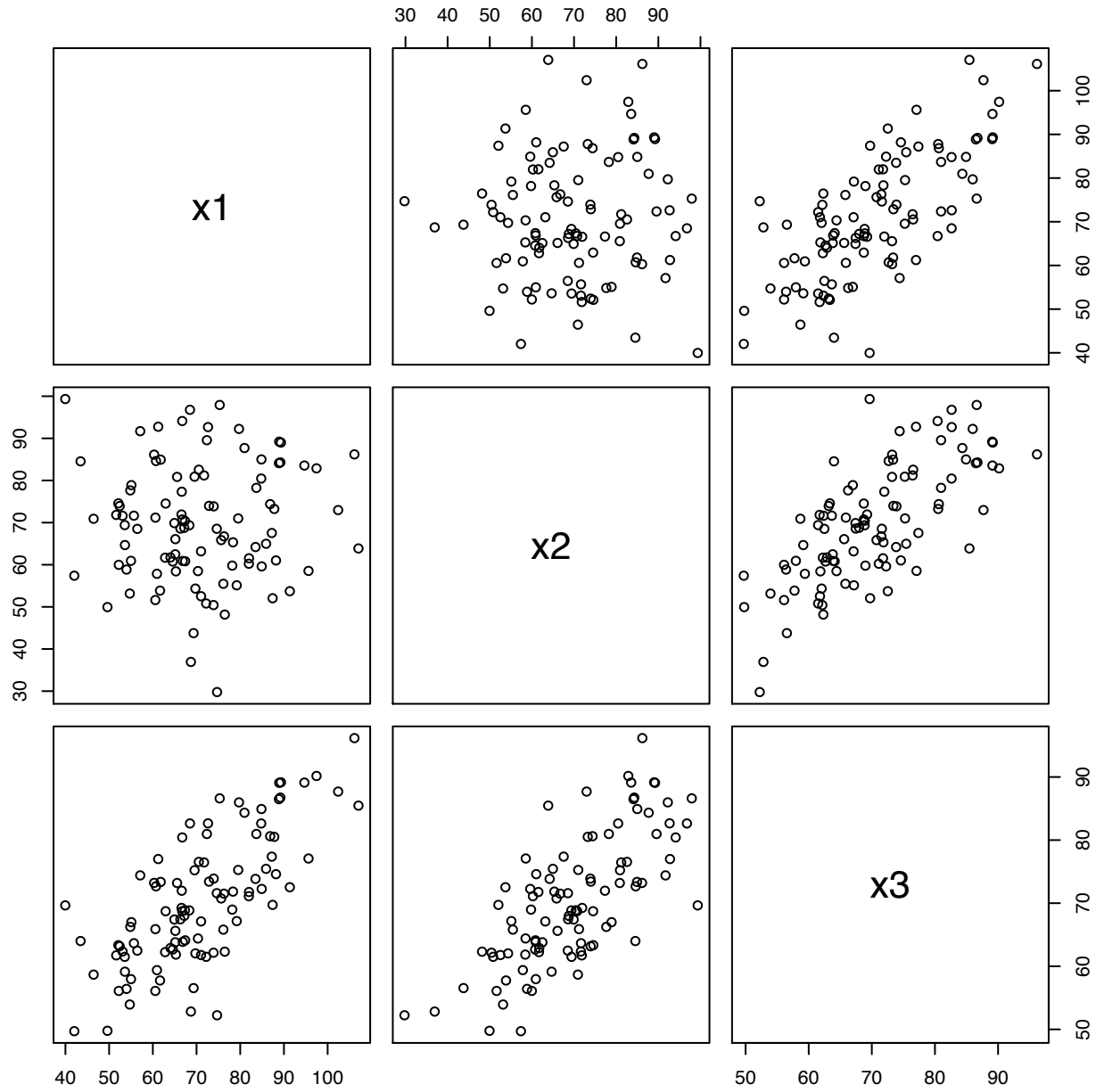

FIGURE 1: *Illustration that a perfect* multi-*collinear relationship might not show up on a pairs plot or in a correlation matrix.*

The ratio between Eqs. 4 and 3 is the **variance inflation factor** for the i^{th} coefficient, VIF_i. The average of the variance inflation factors across all predictors is often written $\overline{VIF}$, or just VIF.

Folklore says that $VIF_i > 10$ indicates "serious" multicollinearity for the predictor. I have been unable to discover who first proposed this threshold, or what the justification for it is. It is also quite unclear what to do about this. Large variance inflation factors do not, after all, violate any model assumptions.

It can be shown that $VIF_i = 1/(1 - R_i^2)$ where R_i^2 is the R^2 you get by regressing X_i on all the other covariates.

Frankly, I don't think many people use VIF.

3 Matrix Perspective

Let $\mathbf{X}$ be the $n \times q$ design matrix. (Remember that $q = p + 1$.) We call $\mathbf{G} = \mathbf{X}^T\mathbf{X}$ the *Gram Matrix*. You should check the following facts:

1. $\mathbf{G}$ is $q \times q$.
2. $\mathbf{G}$ is symmetric.
3. $\mathbf{G}$ is positive semi-definite. That means that, for any vector $\mathbf{a}$ we have that

$$\mathbf{a}^T\mathbf{G}\mathbf{a} \geq 0.$$

Multicollinearity means that there exists a perfect linear relationship between the columns of $\mathbb{X}$. This means that there is a non-zero vector $\mathbf{a} = (a_1, \ldots, a_q)$ such that $\sum_j a_j X_j = 0$ where X_j is the j^{th} column of $\mathbf{X}$. In other words, there exists $\mathbf{a} \neq (0, \ldots, 0)$ such that $\mathbf{X}\mathbf{a} = 0$. Hence

$$\mathbf{a}^T\mathbf{G}\mathbf{a} = 0. \tag{5}$$

Since $\mathbf{G}$ is a square, symmetric, positive-semidefinite matrix, it has a spectral decomposition (or eigen-decomposition). In other words, there are numbers (eigenvalues) $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_q \geq 0$ and vectors (eigenvectors) $\mathbf{v}_1, \ldots, \mathbf{v}_q$ such that:

1. $\mathbf{G}\mathbf{v}_j = \lambda_j \mathbf{v}_j$.
2. $\mathbf{v}_j^T\mathbf{v}_k = 0$ for $j \neq k$.
3. $\mathbf{v}_j^T\mathbf{v}_j = 1$ for each j.
4. $\mathbf{G} = \sum_j \lambda_j \mathbf{v}_j \mathbf{v}_j^T$.
5. $\mathbf{G} = \mathbf{V}\mathbf{D}\mathbf{V}^T$ where the j^{th} column of $\mathbf{V}$ is $\mathbf{v}_j$ and $\mathbf{D}$ is a diagonal matrix with $\mathbf{D}_{jj} = \lambda_j$.
6. The eigenvectors form a basis: any vector w can be written as $w = \sum_j b_j \mathbf{v}_j$ where $b_j = \mathbf{w}^T\mathbf{v}_j$.

Now if the design matrix is collinear then there is a $\mathbf{a}$ such that $\mathbf{a}^T\mathbf{G}\mathbf{a} = 0$. Now

$$0 = \mathbf{a}^T\mathbf{G}\mathbf{a} = \mathbf{a}^T \sum_j \lambda_j \mathbf{v}_j \mathbf{v}_j^T \mathbf{a} = \sum_j \lambda_j \mathbf{a}^T \mathbf{v}_j \mathbf{v}_j^T \mathbf{a} = \sum_j \lambda_j (\mathbf{a}^T \mathbf{v}_j)^2 \equiv U.$$

If $\lambda_q > 0$ then $\lambda_j > 0$ for all j. We know that $(\mathbf{a}^T\mathbf{v}_j)^2 > 0$ for at least one j. (We know this since $\mathbf{a} = \sum_j (\mathbf{a}^T\mathbf{v}_j)\mathbf{v}_j$ and since $\mathbf{a} \neq 0$.) So if $\lambda_q > 0$ then we get $0 = U > 0$ which is a contradiction. We conclude that $\lambda_q = 0$. (There could be other eigenvalues that are 0 as well.)

We have shown that:

Multicollinearity $\implies$ $\mathbf{a}^T\mathbf{G}\mathbf{a} = 0$ for some $\mathbf{a} \neq 0$ $\implies$ at least one eigenvalue of $\mathbf{G}$ is 0.

It is not hard to show that the reverse implications also hold.

3.1 Finding the Eigendecomposition

Because finding eigenvalues and eigenvectors of matrices is so useful for so many situations, mathematicians and computer scientists have devoted incredible efforts over the last two hundred years to fact, precise algorithms for computing them. This is not the place to go over how those algorithms work; it is the place to say that much of the fruit of those centuries of effort is embodied in the linear algebra packages R uses. Thus, when you call

```
eigen(A)
```

you get back a list, containing the eigenvalues of the matrix `A` (in a vector), and its eigenvectors (in a matrix), and this is both a very fast and a very reliable calculation. If your matrix has very special structure (e.g., it's sparse, meaning almost all its entries are zero), there are more specialized packages adapted to your needs, but we don't pursue this further here; for most data-analytic purposes, ordinary `eigen` will do.

3.2 Example

```
> n = 100
> x1 = rnorm(n)
> x2 = rnorm(n)
> x3 = (x1+x2)/2
> y = 5 + 2*x1 + 4*x2 + rnorm(n)
> out = lm(y ~ x1 + x2 + x3)
> summary(out)


Coefficients: (1 not defined because of singularities)
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.14771    0.09667   53.25   <2e-16 ***
x1           1.98474    0.09544   20.80   <2e-16 ***
x2           3.97844    0.08854   44.93   <2e-16 ***
x3                NA         NA      NA       NA


> one = rep(1,n)
> X = cbind(one,x1,x2,x3)
> G = t(X) %*% X
> tmp = eigen(G,symmetric=TRUE)
```

```
> names(tmp)
[1] "values"  "vectors"
> round(tmp$values,5)
[1] 194.00958 100.09923  95.29363   0.00000
>
> out = lm(y ~ x1 + x2)
> summary(out)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.14771    0.09667   53.25   <2e-16 ***
x1           1.98474    0.09544   20.80   <2e-16 ***
x2           3.97844    0.08854   44.93   <2e-16 ***


> X = cbind(one,x1,x2)
> G = t(X) %*% X
> tmp = eigen(G,symmetric=TRUE)
> round(tmp$values,5)
[1] 131.75456  99.98923  93.64998
```

4 Ridge Regression

The real problem with collinearity is that when it happens, there isn't a *unique* solution to the estimating equations. There are rather infinitely many solutions, which all give the minimum mean squared error. This causes the variance of $\widehat{\beta}$ to be infinite.

One solution (which will also help us with high-dimensional regression) is called *ridge regression*. Instead of minimizing

$$\frac{1}{n}(\mathbf{Y} - \mathbf{X}\mathbf{b})^T(\mathbf{Y} - \mathbf{X}\mathbf{b})$$

we instead minimize the penalized squared error

$$\frac{1}{n}(\mathbf{Y} - \mathbf{X}\mathbf{b})^T(\mathbf{Y} - \mathbf{X}\mathbf{b}) + \frac{\lambda}{n}\|\mathbf{b}\|^2.$$

The **penalty factor** $\lambda > 0$ will lead to a solution with some bias but it reduces the variance. In particular, it solves the problem of non-invertibility. We'll come back later to how to pick λ. Setting the gradient to zero at the optimum, $\widehat{\beta}_\lambda$,

$$\mathbf{X}^T\mathbf{Y} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})\widehat{\beta}_\lambda$$

and solve to get

$$\widehat{\beta}_\lambda = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y}.$$

The inverse always exists.

Lecture 21b: Collinearity & Outliers

An **outlier** is a point with a large residual. An **influential point** is a point that has a large impact on the regression. Surprisingly, these are not the same thing. A point can be an outlier without being influential. A point can be influential without being an outlier. A point can be both or neither.

Figure 1 shows four famous datasets due to Frank Anscombe. If you run least squares on each dataset you will get the same output:

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.0001     1.1247   2.667  0.02573 *
x             0.5001     0.1179   4.241  0.00217 **

Residual standard error: 1.237 on 9 degrees of freedom
Multiple R-squared:  0.6665,Adjusted R-squared:  0.6295
F-statistic: 17.99 on 1 and 9 DF,  p-value: 0.00217
```

The top left plot has no problems. The top right plot shows a non-linear pattern. The bottom left plot has an an outlier. The bottom right plot has an influential point. Imagine what would happen if we deleted the rightmost point. If you looked at residual plots, you would see problems in the second and third case. But the resdual plot for the fourth example would look fine. You can't see influence in the usual residual plot.

1 Modified Residuals

Let $\mathbf{e}$ be the vector of residuals. Recall that

$$\mathbf{e} = (\mathbf{I} - \mathbf{H})\epsilon, \quad \mathbf{E}[\mathbf{e}] = \mathbf{0}, \quad \text{Var}(\mathbf{e}) = \sigma^2(\mathbf{I} - \mathbf{H}).$$

Thus the standard error of e_i is $\widehat{\sigma}\sqrt{1 - h_{ii}}$ where $h_{ii} \equiv \mathbf{H}_{ii}$. We then call

$$r_i = \frac{e_i}{\widehat{\sigma}\sqrt{1 - h_{ii}}}$$

the **standardized residual**.

There is another type of residual t_i which goes under various names: the **jackknife residual**, **the cross-validated residual**, **externally studentized residual** or **studentized deleted residual**. Let $\widehat{m}_{i(-i)}$ is the predicted value for the i^{th} data point when (X_i, Y_i) is omitted from the data. Then t_i is defined by

$$t_i = \frac{Y_i - \widehat{m}_{i(-i)}}{s_i} \tag{1}$$

where s_i^2 is the estimated variance of $Y_i - \widehat{Y}_{i(-i)}$. It can be shown that

$$t_i = r_i \sqrt{\frac{n - p - 2}{n - p - 1 - r_i^2}} = \frac{e_i}{\widehat{\sigma}_{(-i)}\sqrt{1 - h_{ii}}} \tag{2}$$

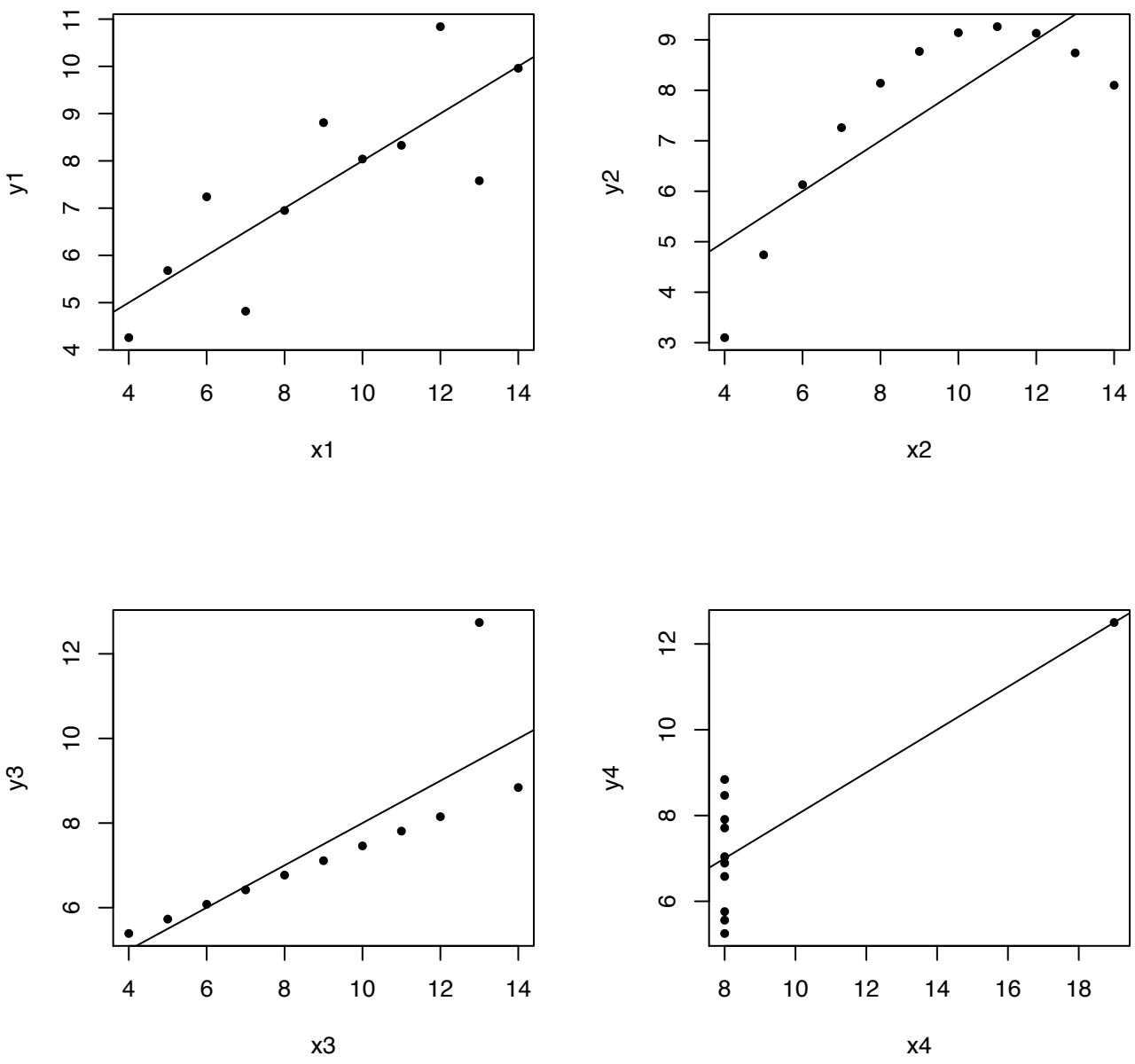

FIGURE 1: *For data sets that have the same fitted line. Top left: no problems. Top right: a non-linear pattern. Bottom left: An outlier. Bottom right: an influential point.*

$\widehat{\sigma}^2_{(-i)}$ is the estimated variance after omitting (X_i, Y_i) is omitted from the data. The cool think is that we can compute t_i without ever having to actually delete the observation and re-fit the model.

Everything you have done so far with residuals can also be done with standardized or jackknife residuals.

2 Influence

Recall that

$$\widehat{\mathbf{m}} = \mathbf{HY}$$

where $\mathbf{H}$ is the hat matrix. This means that each $\widehat{m}_i$ is a linear combination of elements of $\mathbf{H}$. In particular, $\mathbf{H}_{ii}$ is the contribution of the i^{th} data point to $\widehat{m}_i$. For this reason we call $h_{ii} \equiv \mathbf{H}_{ii}$ the *leverage*.

To get a better idea of how influential the i^{th} data point is, we could ask: how much do the fitted values change if we omit an observation? Let $\widehat{\mathbf{m}}^{(-i)}$ be the vector of fitted values when we remove observation i. Then **Cook's distance** is defined by

$$D_i = \frac{(\widehat{\mathbf{m}} - \widehat{\mathbf{m}}^{(-i)})^T(\widehat{\mathbf{m}} - \widehat{\mathbf{m}}^{(-i)})}{(p+1)\widehat{\sigma}^2}.$$

It turns out that there is a handy formula for computing D_i, namely:

$$D_i = \left(\frac{r_i^2}{p+1}\right)\left(\frac{h_{ii}}{1-h_{ii}}\right).$$

This means that the influence of a point is determined by both its residual and its leverage. Often, people interpret $D_i > 1$ as an influential point.

The leave-one-out idea can also be applied to the coefficients. Write $\widehat{\beta}^{(-i)}$ for the vector of coefficients we get when we drop the i^{th} data point. One can show that

$$\widehat{\beta}^{(-i)} = \widehat{\beta} - \frac{(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}_i^T e_i}{1-h_{ii}}. \tag{3}$$

Cook's distance can actually be computed from this, since the change in the vector of fitted values is $\mathbf{x}(\widehat{\beta}^{(-i)} - \widehat{\beta})$, so

$$D_i = \frac{(\widehat{\beta}^{(-i)} - \widehat{\beta})^T\mathbf{X}^T\mathbf{X}(\widehat{\beta}^{(-i)} - \widehat{\beta})}{(p+1)\widehat{\sigma}^2}. \tag{4}$$

Sometimes, whole clusters of nearby points might be potential outliers. In such cases, removing just one of them might change the model very little, while removing them all might change it a great deal. Unfortunately there are $\binom{n}{k} = O(n^k)$ groups of k points you could consider deleting at once, so while looking at all leave-one-out results is feasible, looking at all leave-two- or leave-ten-out results is not.

3 Diagnostics in Practice

We have three ways of looking at whether points are outliers:

1. We can look at their leverage, which depends only on the value of the predictors.

2. We can look at their studentized residuals, either ordinary or cross-validated, which depend on how far they are from the regression line.

3. We can look at their Cook's statistics, which say how much removing each point shifts all the fitted values; it depends on the product of leverage and residuals.

The model assumptions don't put any limit on how big the leverage can get (just that it's ≤ 1 at each point) or on how its distributed across the points (just that it's got to add up to $p+1$). Having most of the leverage in a few super-inferential points doesn't break the model, exactly, but it should make us worry.

The model assumptions *do* say how the studentized residuals should be distributed. In particular, the cross-validated studentized residuals should follow a t distribution. This is something we can test, either for specific points which we're worried about (say because they showed up on our diagnostic plots), or across all the points.

3.1 In R

Almost everything we've talked — leverages, studentized residuals, Cook's statistics — can be calculated using the `influence` function. However, there are more user-friendly functions which call that in turn, and are probably better to use. Leverages come from the 'hatvalues' function, or from the 'hat' component of what 'influence' returns:

```
out = lm(Mobility ~ Commute,data=mobility)
hatvalues(out)
influence(out)$hat  ### this is the same as the previous line
rstandard(out)      ### standardized residuals
rstudent(out)       ### jackknife residuals
cooks.distance(out) ### Cook's distance
```

Often the most useful thing to do with these is to plot them, and look at the most extreme points. The standardized and studentized residuals can also be put into our usual diagnostic plots, since they should average to zero and have constant variance when plotted against the fitted values or the predictors.

```
par(mfrow=c(2,2))
n = nrow(mobility)
out = lm(Mobility ~ Commute,data=mobility)
plot(hatvalue(out),ylab="Leverage")
plot(rstandard(out),ylab="Standardized Residuals")
plot(rstudent(out),ylab="Cross-Validated Residuals")
abline(h=qt(0.025,df=n-2,col="red")
abline(h=qt(1-0.025,df=n-2,col="red")
plot(cooks.distance(out),ylab="Cook's Distance")
```

We can now look at exactly which points have the extreme values, say the 10 most extreme residuals, or largest Cook's statistics:

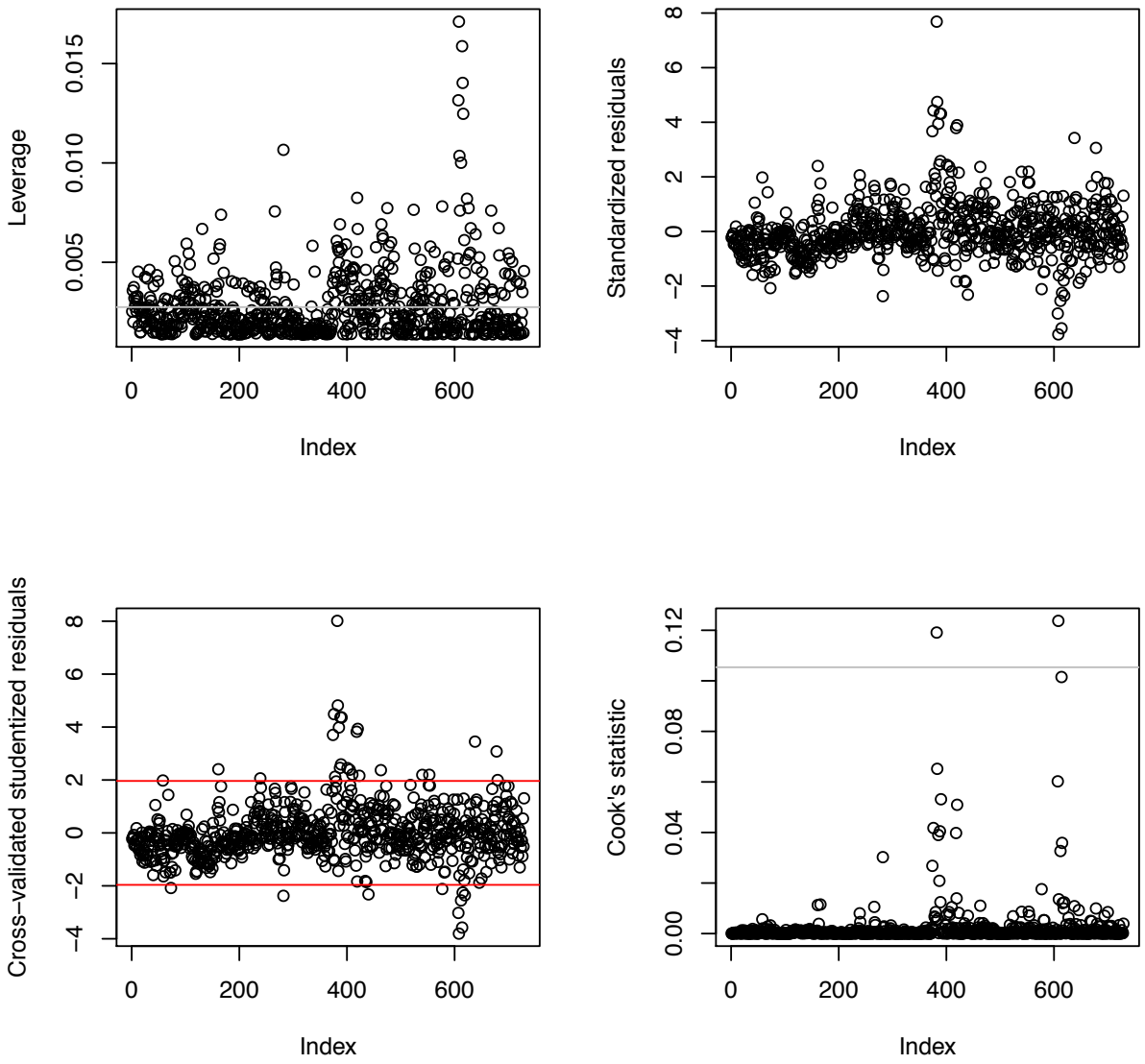

FIGURE 2: *Leverages, two sorts of standardized residuals, and Cook's distance statistic for each point in a basic linear model of economic mobility as a function of the fraction of workers with short commutes. The horizontal line in the plot of leverages shows the average leverage. The lines in studentized residual plot shows a 95% t-distribution sampling interval. Note the clustering of extreme residuals* and *leverage around row 600, and another cluster of points with extreme residuals around row 400.*

```
n = nrow(mobility)
out = lm(Mobility ~ Commute,data=mobility)
r = rstudent(out)
I = (1:n)[rank(-abs(r) <= 10)]   ## indices 10 largest residuals
mobility[I,]
```

```
##       X       Name   Mobility State Commute  Longitude Latitude
## 374 375     Linton 0.29891303    ND   0.646 -100.16075 46.31258
## 376 378 Carrington 0.33333334    ND   0.656  -98.86684 47.59698
## 382 384     Bowman 0.46969697    ND   0.648 -103.42526 46.33993
## 383 385     Lemmon 0.35714287    ND   0.704 -102.42011 45.96558
## 385 388 Plentywood 0.31818181    MT   0.681 -104.65381 48.64743
## 388 391  Dickinson 0.32920793    ND   0.659 -102.61354 47.32696
## 390 393  Williston 0.33830845    ND   0.702 -103.33987 48.25441
## 418 422     Miller 0.31506848    SD   0.697  -99.27758 44.53313
## 420 424 Gettysburg 0.32653061    SD   0.729 -100.19547 45.05100
## 608 618       Nome 0.04678363    AK   0.928 -162.03012 64.47514
```

```
C = cooks.distance(out)
I = (1:n)[rank(-abs(C) <= 10)]   ## indices 10 largest Cook's distances
mobility[I,]
```

```
##       X       Name   Mobility State Commute  Longitude Latitude
## 376 378 Carrington 0.33333334    ND   0.656  -98.86684 47.59698
## 382 384     Bowman 0.46969697    ND   0.648 -103.42526 46.33993
## 383 385     Lemmon 0.35714287    ND   0.704 -102.42011 45.96558
## 388 391  Dickinson 0.32920793    ND   0.659 -102.61354 47.32696
## 390 393  Williston 0.33830845    ND   0.702 -103.33987 48.25441
## 418 422     Miller 0.31506848    SD   0.697  -99.27758 44.53313
## 420 424 Gettysburg 0.32653061    SD   0.729 -100.19547 45.05100
## 607 617   Kotzebue 0.06451613    AK   0.864 -159.43781 67.02818
## 608 618       Nome 0.04678363    AK   0.928 -162.03012 64.47514
## 614 624     Bethel 0.05186386    AK   0.909 -158.38213 61.37712
```

4 Dealing With Outliers

There are essentially three things to do when we're convinced there are outliers: delete them; change the model; or change how we estimate. None of these should ever be done lightly, and it is crucial to be transparent throughout the process. One should never throw out data without specifically making a note of it and explaining the justification.

4.1 Deletion

The best case for removing a data point is if you have good reasons to think that the data point belongs to a different phenomenon or population from the one you're studying. (You're trying to see if a new drug helps cancer patients, but you discover the hospital has included some burn patients and influenza cases as well.) Or the data point does belong to the right population, but also somehow to another one which isn't what you're interested in right now. (All of the data is on cancer patients, but some of them were also sick with the flu.) You should be careful about that last, though. (After all, some proportion of future cancer patients are also going to have the flu.)

The next best scenario after that is that there's nothing quite so definitely wrong about the data point, but it just looks really weird compared to all the others. Here you are really making a judgment call that either the data really are mistaken, or not from the right population, but you can't put your finger on a concrete reason why. The rules-of-thumb used to identify outliers, like "Cook's distance shouldn't be too big", or "Tukey's rule" which flags any point more than 1.5 times the inter-quartile range above the third quartile, or below the first quartile. It is always more satisfying, and more reliable, if investigating how the data were gathered lets you turn cases of this sort into one of the two previous kinds.

The least good case for getting rid of data points which isn't just bogus is that you've got a model which almost works, and would work a lot better if you just get rid of a few stubborn points. This is really a sub-case of the previous one, with added special pleading on behalf of your favorite model. You are here basically trusting your model more than your data, so it had better be either a really good model or really bad data.

4.2 Changing the Model

Outliers are points that break a pattern. This can be because the points are bad, or because we made a bad guess about the pattern. For example, data from a quadratic regression will be definite outliers for any linear model. Deleting them, in order to make a linear model work better, would be short-sighted at best.

The moral is that data points can look like outliers because we're looking for the wrong pattern. If when we find apparent outliers and we can't convince ourselves that data is erroneous or irrelevant, we should consider changing our model, before, or as well as, deleting them.

4.3 Robust Linear Regression

A final alternative is to change how we estimate our model. Everything we've done has been based on ordinary least-squares (OLS) estimation. Because the squared error grows very rapidly with the error, OLS can be very strongly influenced by a few large residuals. We might, therefore, a different method of estimating the parameters, e.g., minimizing the sum of absolute instead of squared errors. Estimation techniques which are less influenced by outliers in the residuals than OLS are called **robust estimators**, or (for regression models) **robust regression**. We may discuss these in detail later in class.